Using The GSM/UMTS SIM to Secure Web Services

John A. MacDonald*
Information Security Group
Royal Holloway, University of London,
Egham, England TW20 0EX
john@madgo.com

Chris J. Mitchell
Information Security Group
Royal Holloway, University of London,
Egham, Surrey TW20 0EX, UK
c.mitchell@rhul.ac.uk

Abstract

In this paper we present a Mobile Operator endorsed authentication and payment platform for the consumption of web services by a Mobile Station. We propose a protocol where the Mobile Operator plays the role of Trusted Third Party to issue authentication and authenticated payment authorisation tokens to facilitate a transaction between a Mobile Station and a Web Service Provider. We consider the structure and syntax of these tokens to minimise service latency, and provide security services to protect against the threat model. To validate our proposal we have developed code to create a Web Service test scenario utilising readily available J2ME, Java Card, J2SE and J2EE platforms, Web Services tools from Apache, the KToolBar emulator from Sun, and a Gemplus Java Card.

1. Introduction

This paper proposes a protocol for authentication and payment between a consumer and a Web Service Provider that builds upon the Mobile Operator relationship with the mobile subscriber. The protocol enables the Mobile Operator to provide a secure and trusted payment and authentication environment, allowing Web Service Providers to gain commercial access to the Mobile Operator's subscriber base.

But why should the Mobile Operator wish to encourage such access? It has long been noted [4] that distribution structures, and specifically the consumer facing retailing function, evolve as industries mature. Many consider traditional Mobile Operators to be at the early stages of their development as retailers of digital content. The current distribution structures typified by Vodafone Live! from Vodafone, T Zones from T-Mobile, and e-mocion from Telefonica are examples of "one stop shops". Vertically integrated, they source, market and advertise a range of goods to consumers who are encouraged to repeat purchase. They may be considered as analogous to a Department Store on the high street. The typical High Street has evolved, however, and in many cases is complemented (if not replaced) by the Shopping Mall. Comprising both Department Stores and specialist retailers the operator of the Shopping Mall benefits from a large number of customers (i.e. traffic volume) whilst remaining independent from the cost and management of the retailed stock. As the commercial benefit from provision of digital content to mobile consumers transitions from promotional to revenue generating, the "Shopping Mall" concept of digital content retailing may become an attractive model for the traditional Mobile Operator.

This paper focuses on two critical enablers facilitating the transition of Mobile Operator digital content retailing functions to a services orientated architecture [10], namely authentication and payment.

2. The web service requirement

Our proposal concerns three main actors; the Consumer, the Mobile Operator and the Service Provider.

The consumer is assumed to access the scheme via a bandwidth-constrained Mobile Station, comprising mobile device and service-enabling SIM card connected to a GPRS or UMTS mobile network. Service latency should be minimal without the need to purchase new equipment, and the "purchase experience" should be consistent across all services, irrespective of the actual service provider. Payment for services should be through the normal mechanisms provided by the Mobile Operator, with anonymity an optional consumer requirement. Service consumption is ad hoc, irregular and transitory in duration.

* This work was supported by sponsorship funding from Telefonica Móviles, España.

The Mobile Operator is assumed to require maximum distribution for the available services at lowest cost. Consumers and Service Providers should be capable of dynamically and asynchronously entering and leaving the system. The service must be terminal vendor independent and capable of being set-up using Over The Air (OTA) techniques. Finally, the Service Provider, who will not want to develop new business processes solely for Mobile Operators, must interface to the system using standard internet protocols.

We base our proposed scheme on the assumption that these requirements are met with a Web Services architecture, where:

- the consumer service endpoint is an OTA installed application running on a mobile device that uses the SIM card as its security element,
- the service content is provided by a Web Services Provider in accordance with internet standards, and
- the Mobile Operator provides service discovery, content delivery, authentication and payment services to both consumers and Web Service Providers.

3. The proposed scheme

In our proposal the Mobile Operator is a trusted third party acting as an intermediary between Web Service Providers and each of the Web Service consumers (Mobile Stations) served by the Mobile Operator. The Mobile Operator provides authentication and payment authorisation tokens, that are exchanged by the Mobile Station for the subsequent consumption of Web Services provided by Web Service Providers that are listed by the Mobile Operator.

The Mobile Station is assumed to implement a *Security Agent* function — an example of which is presented in [8]. The *Security Agent* comprises a device executed MIDlet application for I/O and computationally intensive operations, together with a tamper-resistant module (e.g. Trusted Programme Module (TPM) and/or SIM card) executed application for secure storage and cryptographic processing. The Mobile Operator is assumed to implement a Token Distribution Centre comprising Authentication and Payment Servers working in consert with the Billing System.

We adopt a push-based model [5] to exchange authentication and payment authorisation tokens between the scheme entities. Tokens are pushed from the Mobile Operator's Token Distribution Centre to the Mobile Station, for local storage. This allows a shopping basket of services to be assembled before the tokens are subsequently pushed from the Mobile Station to the Web Service Providers in exchange for their services.

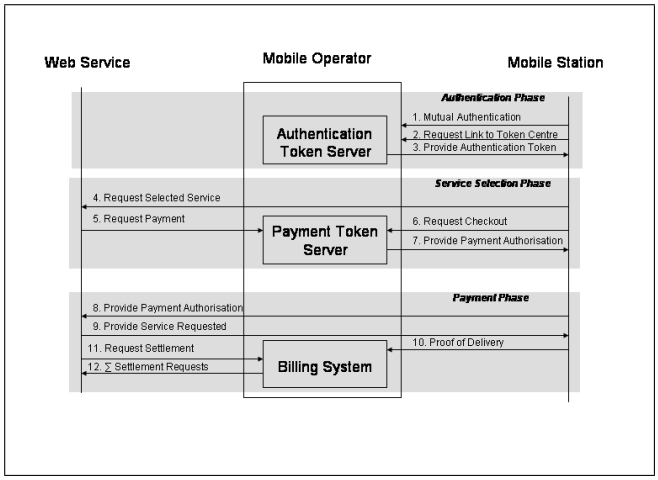

Figure 1. Scheme Description

By storing the tokens on the Mobile Station we simulate a familiar shopping behaviour. We allow the consumer to pause (i.e. service interuption) between the phases of entering the Shopping Mall (i.e. authentication), browsing and selecting the goods (web service selection) and proceeding to the checkout (i.e. payment) as indicated in the shaded areas of Figure 1. It is considered good practice [2] to design mobile applications so that they can be interrupted by the user. The scheme is summarised with reference to Figure 1.

1. The *Security Agent* of the Mobile Station and the Mobile Operator perform a mutual authentication process, using the technique described in [8]. This initiates a secure session, including the agreement of cryptographic algorithms and the establishment of a pair of authenticated shared secret keys, one for encryption (CK) and the other for MAC generation (IK).
2. The Mobile Station application requests a link to the Mobile Operator's Token Distribution Centre. A mobile identifier used solely for web services is provided, and an authentication token requested. For improved privacy, this web services mobile identifier can be encrypted using the confidentiality key CK if required.
3. The Authentication Server issues an authentication token to the Mobile Station.
4. The Mobile Station application browses the web services offered by the Mobile Operator. Upon ser-

vice selection, the authentication token is provided to the Web Service Provider as proof of identity.

5. The Web Service Provider responds by informing the Mobile Operator's Payment Token Server of the price and other contractual terms of the service requested by the Mobile Station.
6. Once service selection is completed, the Mobile Station requests checkout.
7. The Payment Token Server responds by issuing a single payment token to the Mobile Station for all the selected services.
8. To initiate service consumption the Mobile Station pushes the one-time-use payment token to the appropriate Web Service Provider(s). To avoid information leakage between Web Service Providers, the Mobile Station can request a Payment token for each service selected. In this instance service consumption is initiated by the Mobile Station pushing the one-time-use payment token only to the specific Web Service Provider.
9. The Web Service(s) responds by supplying the requested service to the Mobile Station via the Mobile Operator. If required the service may be encrypted using CK to provide confidentiality services between the Mobile Operator and the Mobile Station.
10. Proof of delivery notification is provided by the Mobile Station to the Billing System, and the secure session is released.
11. Financial settlement request is made by the Service Provider to the Mobile Operator, as determined by their commercial agreement.
12. Settlement between Mobile Operator and Web Service provider is made either on a per transaction or per time period basis. Settlement between Mobile Station and Mobile Operator can either be on a prepay basis, occurring when the payment token is issued in step 7 above, or upon receipt of proof-of-delivery.

4. Implementation options

Web Services are defined [13] as software systems that support interoperable network interactions. They allow implementation of a service-orientated architecture incorporating the entities of Service Provider, Service Consumer and Service Registry. For information to be moved around the network it must be packaged in a format that is understood by these entities. The Simple Object Access Protocol, (SOAP) [13] is the standardised packaging protocol currently utilised for Web Services. SOAP supports information exchanges by specifying a way to structure XML messages.

As in any open network environment, these exchanges are exposed to security threats of message leakage, tampering and vandalism. We propose protocol and token implementation options that are designed to resist masquerading, message tampering, replay, and denial of service attacks. Further, as the characteristic of a Web Service is a response to a message, perceived service quality is also dependent on latency between message and response. We therefore also consider the implementation options that affect this.

We present both specific protocol exchanges and the structure and syntax of the authentication and payment tokens.

4.1. Prerequisites for protocol

Our protocol uses both symmetric and asymmetric cryptographic techniques to provide the authentication and integrity services required.

We choose to use symmetric rather than asymmetric cryptography for providing the security services between Mobile Station and Mobile Operator. Performance is critical in a mobile system and overhead must always be minimised wherever possible [1]. A long term secret key K_{SC} is shared by the Mobile Operator and the Mobile Station's SIM card. This is used as described in [8] for mutual authentication and authenticated key establishment.

By contrast we use asymmetric cryptography to provide the security services between the Mobile Operator and the Web Service Provider. Unlike the one-to-one and enduring relationship between Mobile Station and Mobile Operator, our scheme assumes that a Web Service Provider may have a transitory relationship with multiple Mobile Operators. In this topology it is best to avoid the necessity of establishing a long term shared secret; we thus adopt an asymmetric cryptographic solution for provision of security services. The Mobile Operator generates asymmetric key pairs for the Authentication Server and the Payment Server, and obtains certificates Cert_{OP_AS} and Cert_{OP_PS} for the respective public keys from a Certification Authority. Likewise, the Web Service Provider generates a key pair for its Web Service Server, and obtains certificate Cert_{WS} for the public key. The private keys will be used for digitally signing messages. Our protocol is based on the assumption that the Web Service Provider has access to a trusted copy of the public key of the Certification Authority used to sign the Mobile Operator's pub-

lic key certificates, and vice versa. We also assume that the Mobile Operator's Authentication Server certificate Cert_{OP_AS} and the Mobile Operator Payment Server certificate Cert_{OP_PS} are in a format processable by the Web Service Provider, and that the Web Server Provider certificate Cert_{WS} is in a format processable by the Mobile Operator. It is further assumed that the Web Service has access to the Mobile Operator's certificates, prior to commencement of the protocol. In a multiple Mobile Operator topology the certificates for each specific Mobile Operator may be accessed via the Mobile Operator specific value of the `SecurityDomain` element of the authentication token as presented in appendix 1. The associated trust issues are outside the scope of this paper.

The Mobile Station identifier i_M is a unique identifier assigned by the Mobile Operator, used specifically for procuring web services and distinct from the IMSI. Its broadcast must be minimised wherever possible to maintain user privacy and reduce the threat of cloning. Therefore i_M is only transmitted once to identify the Mobile Station to the Mobile Operator. Following authentication, a temporary Mobile Station identifier $i_{M'}$ is used to identify the Mobile Station in all communications with the Web Service Provider. This ensures user anonymity and reduces the risk of cloning. It is the Mobile Operator's responsibility to maintain the relationship between i_M and $i_{M'}$.

4.2. Protocol

We describe the critical protocol exchanges to address the threat model by considering the authentication, service selection and payment phases of the protocol. Our description assumes that an authenticated key establishment process has taken place between the Mobile Operator and the *Security Agent* of a Mobile Station [8].

We adopt the following additional notation:

$$\begin{aligned} AS &= \text{Authentication Server} \\ PS &= \text{Payment Server} \\ MS &= \text{Mobile Station} \\ WS &= \text{Web Service} \\ S(D, s_x) &= \text{Signature of data } D \text{ using key } s_x \\ \text{MAC}_{IK}(D) &= \text{MAC on data } D \text{ using key } IK \\ S_{i_{M'}} &= \text{SAML authentication assertion} \\ P_{i_{M'}} &= \text{authenticated payment authorisation} \end{aligned}$$

4.2.1. *Authentication:* Mobile Station requests an authentication token. The Authentication phase commences with the authenticated key establishment process. The Mobile Station then requests an authentication token from the Mobile Operator's Authentication Server by sending its identity i_M integrity protected with a MAC computed with the Mobile Operator and *Security Agent* shared secret integrity key IK.

$$MS \rightarrow AS:\ i_M \| \text{MAC}_{IK}(i_M)$$

Successful verification of the MAC confirms that the Mobile Station at the other end of the secure channel is the legitimate owner of the mobile identity i_M. Note that freshness is guaranteed by the fact that the key IK used to generate the MAC is session-specific. The Authentication Server generates a temporary mobile identifier $i_{M'}$ and records the mapping between i_M and $i_{M'}$. The server then creates a SAML authentication assertion $S_{i_{M'}}$ using the temporary Mobile Station identifier $i_{M'}$. This assertion is compliant with the SAML [9] specification and confirms that the user was authenticated at a specific time by the Mobile Operator's Authentication Server. This is digitally signed by the private key s_{OP_AS} corresponding to the public key certificate Cert_{OP_AS} to give the authentication token $S_{i_{M'}} \| S(S_{i_{M'}}, s_{OP\_AS})$. The authentication token is then integrity protected with a MAC computed using the shared secret IK and transferred to the Mobile Station.

$$\begin{aligned} AS \rightarrow MS:\quad & S_{i_{M'}} \| S(S_{i_{M'}}, s_{OP\_AS}) \| \\ & \text{MAC}_{IK}(S_{i_{M'}} \| S(S_{i_{M'}}, s_{OP\_AS})) \end{aligned}$$

After successful verification of the MAC by the *Security Agent* using IK, the authentication token $S_{i_{M'}} \| S(S_{i_{M'}}, s_{OP\_AS})$ is stored in persistent Mobile Station memory.

4.2.2. *Service Selection:* Mobile Station receives payment token. The Mobile Station browses the Mobile Operator's selection of web services. Upon service selection, the Mobile Station copies the authentication token $S_{i_{M'}} \| S(S_{i_{M'}}, s_{OP\_AS})$ from persistent memory and transfers it to the Web Service Provider as proof of identity.

$$MS \rightarrow WS:\ S_{i_{M'}} \| S(S_{i_{M'}}, s_{OP\_AS})$$

The Web Service can determine the identity of the Mobile Operator from the `SecurityDomain` element in the `AuthenticationAssertion` contained within $S_{i_{M'}}$ (see Appendix 1). Assuming contractual terms have been agreed with the specific Mobile Operator for provision of the requested service, then the Web Service verifies the digital signature using the public key in Cert_{OP_AS}, which it is assumed that the Web Service application has the means to verify. To verify the freshness of the authentication token, the web service application checks the timestamp attribute `IssueInstant` of element `AuthenticationAssertion` within $S_{i_{M'}}$,

and also retains a log of recently accepted tokens to ensure that no token is accepted twice. Following successful verification of the digital signature and freshness checks, the Web Service Provider responds by informing the Mobile Operator's Payment Token Server of service identity, payment details and other contractual terms, T_{ws}, requested by the authenticated entity identified in authentication assertion $S_{i_{M'}}$. The authentication assertion and the terms of the selected service are digitally signed using the Web Server's private key corresponding to the public key in Cert_{WS}, and are transferred to the Mobile Operator's Payment Server.

$$WS \rightarrow PS: \ S_{i_{M'}} \| T_{ws} \| S(S_{i_{M'}} \| T_{ws}, s_{WS}) \| \text{Cert}_{WS}$$

The Mobile Operator's Payment Server verifies the digital signature using the public key in Cert_{WS}. Following successful verification, the identity and contractual terms of the web service described in T_{ws}, are stored for subsequent checkout against the Mobile Station identity i_M. This occurs for every service selected by the Mobile Station. Once service selection is completed, the Payment Server confirms the user's credit worthiness, reserves the total payment amount, and creates a payment authorisation $P_{i_{M'}}$ for all selected services. This is digitally signed using the Mobile Operator's private key corresponding to the public key in Cert_{OP_PS}, to yield the authenticated payment token $P_{i_{M'}} \| S(P_{i_{M'}}, s_{OP\_PS})$. The payment token is then integrity protected with a MAC computed using the shared secret IK, and transferred to the Mobile Station.

$$PS \rightarrow MS: \quad P_{i_{M'}} \| S(P_{i_{M'}}, s_{OP\_PS}) \| \text{MAC}_{IK}(P_{i_{M'}} \| S(P_{i_{M'}}, s_{OP\_PS})$$

After successful verification of the MAC by the *Security Agent* using IK, the authenticated payment token $P_{i_{M'}} \| S(P_{i_{M'}}, s_{OP\_PS})$ is stored in persistent Mobile Station memory by the *Security Agent*.

4.2.3. *Payment:* Mobile Station exchanges token for services. To initiate consumption of the selected services, the Mobile Station copies from persistent memory the authenticated payment token $P_{i_{M'}} \| S(P_{i_{M'}}, s_{OP\_PS})$ and transfers it to the Web Service Provider(s).

$$MS \rightarrow WS: \ P_{i_{M'}} \| S(P_{i_{M'}}, s_{OP\_PS})$$

The Web Service can determine the identity of the Mobile Operator from the `SecurityDomain` element in the `AuthenticationAssertion` contained within $P_{i_{M'}}$ (see Appendix 1). Assuming contractual terms have been agreed with the specific Mobile Operator for provision of the requested service, then the Web Service verifies the digital signature using the public key in Cert_{OP_PS}, which it is assumed that the Web Service application has the means to verify. Replay attacks are avoided by ensuring that payment tokens are one-time-use only. Token freshness is confirmed by the Web Service checking and logging the service identity and the timestamp attribute `IssueInstant` in `AuthenticationAssertion`. Following successful signature verification and token freshness tests the Web Service Provider responds by supplying the service to the Mobile Station.

4.3. Authentication & payment tokens

To maximise interoperability these tokens are designed as XML documents. The $S_{i_{M'}}$ and $P_{i_{M'}}$ documents assert user-related facts about authentication and payment authorisation respectively. For integrity, source entity authentication and non-repudiation these assertions are signed by the issuing authority, i.e. the Mobile Operator. They must therefore include an XML digital signature. Unfortunately the self describing, extensible and interoperable nature of XML leads to redundancy and, as a consequence, potentially increased latency in constrained-bandwidth networks. With web service performance measured in throughput and latency, the use, structure and syntax of the authentication $S_{i_{M'}} \| S(S_{i_{M'}}, s_{OP\_AS})$, and payment $P_{i_{M'}} \| S(P_{i_{M'}}, s_{OP\_PS})$ tokens is a critical implementation choice. We identify the following implementation objectives regarding the use and design of authentication and payment tokens:

1. Minimise the transmission time for $S_{i_{M'}} \| S(S_{i_{M'}}, s_{OP\_AS})$ and $P_{i_{M'}} \| S(P_{i_{M'}}, s_{OP\_PS})$ over bandwidth-constrained networks. Recasting the scheme of Figure 1 into a pull-based model, where the Mobile Operator provides a full proxy service on behalf of the mobile client, eliminates the need to both transmit the tokens over the constrained bandwidth mobile network and to store them locally in expensive client memory. This option is, however, at the expense of more centralised application control by the Mobile Operator.
2. Compress the XML documents $S_{i_{M'}}$ and $P_{i_{M'}}$. There exist various XML compression schemes that exploit the tree structure of the language and enable the removal of unnecessary white space [6]. As document size and latency must be minimised in the proposed scheme, both the compression ratio and the total transfer time (including time to compress, transmit and subsequently decompress) of the compressed document are criti-

cal metrics for the choice of compression engine. Although compression potentially reduces memory requirements and latency, the compressed tokens are non-standard. This could lead to interoperability issues for the scheme, unless compression is only deployed over the bandwidth constrained link. Such an implementation would once again be at the expense of more centralised application control by the Mobile Network. It is worth noting that such a Mobile Network specific XML compression technique, WBXML [15], which employs a scheme whereby the most common occurring XML value is represented by the smallest token, was omitted from version 2 of the OMA DRM Rights Object Acquisition Protocol [12] despite its inclusion in the earlier version 1 [11]. WBXML's inability to compress binary data such as signatures and certificates, and the interoperability concern, demonstrates the practical issues surrounding the choice of implementing XML compression for the tokens $S_{i_{M'}} \| S(S_{i_{M'}}, s_{OP\_AS})$ and $P_{i_{M'}} \| S(P_{i_{M'}}, s_{OP\_PS})$. Adoption of a standard compression technique such as http deflate is proposed as a reasonable compromise to minimise transmission latency whist ensuring maximum interoperability.

3. Optimally design the token syntax. The XML signature syntax standard is designed with a high degree of extensibility and flexibility. To optimise performance of the proposed scheme, it is possible to select an implementation of $S_{i_{M'}} \| S(S_{i_{M'}}, s_{OP\_AS})$ and $P_{i_{M'}} \| S(P_{i_{M'}}, s_{OP\_PS})$ that minimises size and latency whilst remaining compliant to the standard to ensure interoperability. We propose the following implementation choices within the XML digital signature [3].

   (a) To adopt an enveloping XML Signature structure where the authentication and payment tokens are child elements to the signature. This allows the Mobile Station to easily access the token in the `Object` container element to confirm validity before invoking expensive network resources.

   (b) To reduce the size and the speed of the signing operation, each resource within the signature manifest of the `SignedInfo` element is hashed, and the digital signature of the hashed list then calculated.

   (c) To protect against an attack that attempts to substitute the `Object` element of token $P_{i_{M'}}$ and $S_{i_{M'}}$ the canonicalisation algorithm used is referenced in `CanonicalizationMethod`.

   (d) To relieve the limited processing power Mobile Station of the computational expense of validating the digital signatures, tokens $S_{i_{M'}} \| S(S_{i_{M'}}, s_{OP\_AS})$ and $P_{i_{M'}} \| S(P_{i_{M'}}, s_{OP\_PS})$ are transmitted to the Mobile Station via a secure, symmetric integrity protected channel. Maximum interoperability with web services, and the requirement for Web Service Providers to dynamically and asynchronously enter and leave the system without the need for complex secret key distribution structures, is achieved by using a digital signature, not a message authentication code, in element `SignatureMethod`.

   (e) To prevent an attacker removing the assertion and payment tokens from the `Object` element, whilst retaining verification of the digital signature, the `Object ID` URI is referenced in the `Reference` element.

   (f) To reduce the size of the XML documents, the optional element `KeyInfo` is excluded from the XML Signature structure. Key verification material of tokens $S_{i_{M'}} \| S(S_{i_{M'}}, s_{OP\_AS})$ and $P_{i_{M'}} \| S(P_{i_{M'}}, s_{OP\_PS})$ and the issue of trust is left to the web service application. This therefore assumes that the web service application has access to the key verification material. This assumption is valid if the authentication assertion $S_{i_{M'}}$ is based on SAML [9]. As such, the assertion will specify the identity of the authentication authority (i.e. the Mobile Operator). As the web service is listed by the Mobile Operator, it is considered to be within the Mobile Operator's domain, and therefore trust is established prior to the receipt of the tokens $S_{i_{M'}} \| S(S_{i_{M'}}, s_{OP\_AS})$ and $P_{i_{M'}} \| S(P_{i_{M'}}, s_{OP\_PS})$.

   (g) To thwart any fraudulent attempt by the Mobile Station to resubmit a payment token, a timestamp attribute `IssueInstant` is included in the `AuthenticationAssertion` within $S_{i_{M'}}$.The web service application checks that no two payment tokens from the same issuer are redeemed by the same user with the same timestamp. The authenticated payment token authorisation $P_{i_{M'}} \| S(P_{i_{M'}}, s_{OP\_PS})$ contains the ini-

tial authentication assertion $S_{i_{M'}}$. To reduce the storage requirement for long term protection against Mobile Station initiated replay attacks, an element specifying the token's validity period may be added to the authenticated payment token $P_{i_{M'}}$. For simplicity of presentation this element is not presented in appendix 1.

(h) To avoid issuing a payment token for each service requested, at the expense of information leakage as noted above, a single payment token is issued for a shopping basket of services. The services to be purchased are detailed in `SignatureProperties`. The semantics of `SignatureProperties` are application-specific, but would typically contain the supplier identification, item specifier or sku (stock keeping unit) and, perhaps, the price.

This results in the representative syntax for a signed authenticated Payment token P_x, as presented in appendix 1.

4.4. Proof of concept prototype

To validate our proposal we have constructed the Proof of Concept model of Figure 2, based on readily available open source tools:

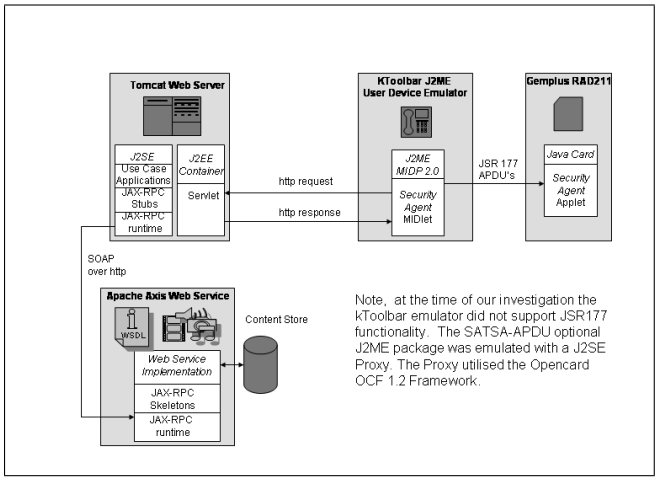

Figure 2. Proof of Concept Implementation

- A J2EE Servlet web application performs the Mobile Operator function and is packaged as a WAR file (Web Application Archive) for easy deployment on a Tomcat Apache Web Server.
- The Mobile Station comprises a mobile device and a SIM card. A J2ME Client performs the mobile device function and is emulated by the Wireless KToolbar [14] from Sun Microsystems, running our *Security Agent* MIDP 2.0 MIDlet on the reference J2ME implementation. The SIM card *Security Agent* function is provided by a Gemplus GemXpresso RAD 211 Java Card with crypto package, connected to our demonstration environment via a USB card reader.
- A Web Service application is packaged as a WAR file and deployed on the Tomcat Server and communicates with the Mobile Operator function using SOAP over http. We used the jax-rpc API together with tools from Apache Axis to create the service WSDL and deploy the Web Service on a Tomcat Server.

The demonstration environment of our proof of concept model is implemented in J2SE. J2SE provides the necessary Java Swing classes for monitoring the various use case applications tested on our model. The model is designed so that each phase of a specific use case is initiated manually and monitored by visual feedback through the use of J2SE's GUI `LayoutManager` class and `ActionListener` interface.

The test scenario is the implementation of a simulated football match, where the consumer selects the specific match venue of interest via a mobile device resident "Football Service" J2ME application. The content provided by the Web Service comprises match data (team identities, current score, match duration, and attendance), graphics (a png image file of a critical match event, e.g. a goal) and a Rights Object controlling consumption of the service. The Web Service Deployment Descriptor and the WSDL of the simulated football match service (MatchCentreService) are presented in appendices 2 and 3 respectively. In our test model the graphic file was encrypted for confidentiality and the Rights Object (`serviceConditions`) accompanying the user content was used to convey content usage constraints; full details are provided in [7].

Invocation of the J2ME "Football Service" application initiates mutual authentication concluding with the creation of a high bandwidth secure channel between Server and SIM card as detailed in [8] and establishment of the shared secret IK. The Server issues authentication and payment tokens, which are exchanged for the Web Service as described above. The content associated with the service is provided to the Mobile Operator using SOAP over http, and then securely transferred to the Mobile Station for consumption via the high bandwidth channel implemented by the J2ME and Java Card *Security Agent*.

5. Conclusion

In this paper we have introduced a scheme for authentication and payment between a consumer and a Web Service Provider that builds upon the Mobile Operator relationship with the mobile subscriber. We have considered the scheme protocol, and the structure and syntax of the payment and authentication tokens in detail. We recommend specific implementation options that optimise performance between the conflicting requirements of interoperability and service latency whilst providing security against external attack. We have modelled our proposal with open source tools based on a Java solution to supply a web service of a simulated football match to a mobile consumer via a Mobile Operator trusted third party.

To summarise, our proposal provides:

1. the user with a high level service discovery interface plus anonymity from Web Service Providers;
2. the Mobile Operator with a pivotal role and a revenue generating opportunity in the provision of a web services security and payment platform;
3. the Content Provider with a secure, scalable distribution channel.

In conclusion, our proposal allows the traditional Mobile Operator to leverage the breadth, innovation and marketing diversity provided by the web services developer community. By adopting such a dynamic, flexible, services orientated architecture we encourage Mobile Operators to rethink their current distribution structures and business models for the sourcing and delivery of digital content to their mobile subscribers.

References

[1] C. W. Blanchard and N. Trask. Wireless security. In R. Temple and J. Regnault, editors, *Internet and Wireless Security*, number 4 in BT Exact Communications Technology Series, chapter 9, pages 146–170. IEE, London, 2002.

[2] Cynthia Block and A. C. Wagner. *MIDP 2.0 Style Guide.* Addison-Wesley, London, 2003.

[3] D. Eastlake, J. Reagle, and D. Solo. *XML-Signature Syntax and Processing.* http://www.w3.org/, 2001.

[4] R. Ford. Managing retail service businesses for the 1990s: Marketing aspects. *European Management Journal*, 8:58–66, 1990.

[5] Sankar Krishna. *Web Services Framework and Assertion exchange using SAML.* W3C, http://www.w3.org, 2001.

[6] Weimin Li. Xcomp: An XML compression tool. MSc thesis, University of Waterloo, Ontario, Canada, 2003.

[7] John A. MacDonald and Chris J. Mitchell. Content centric DRM for mobile vertical market. Information Security Group, Royal Holloway, University of London — Internal paper, November 2004.

[8] John A. MacDonald, William G. Sirett, and Chris J. Mitchell. Overcoming channel bandwidth constraints in secure SIM applications. In *Security and Privacy in the Age of Ubiquitous Computing.* Springer Science and Business Media, 2005.

[9] E. Maler, P. Mishra, and R. Philpott. *Assertions and Protocol for the OASIS Security Assertion Marup Language SAML v1.1.* http://www.oasis-open.org/, 2003.

[10] Richard Monson-Haefel. *J2EE Web Services.* Addison-Wesley, 2004.

[11] OMA-DRMREL-v1. *Rights Expression Language v1.0.* http://www.openmobilealliance.org, 2002.

[12] OMA-DRMREL-v2. *Rights Expression Language v2.0.* http://www.openmobilealliance.org, 2004.

[13] Jason Snell, Doug Tidwell, and Pavel Kulchenko. *Programming Web Services with SOAP.* O'Reilly, 2002.

[14] Sun Microsystems, http://java.sun.com/products. *Wireless Toolkit, Version 2.1,*, 2003.

[15] WAP-192-WBXML. *Binary XML Content Format Specification.* http://www.wapforum.org, 2001.

A. Appendix 1

The representative syntax for a signed authenticated Payment token P_x is presented below. The syntax for the simpler signed authentication token S_x is similar but excludes the `SignatureProperties` element:

```
<Signature xmlns ="http://www.w3.org/2000/09/xmldsig\#"
    xmlns:app ="http://www.madgo.com/simple">
 <SignedInfo>
  <CanonicalizationMethod Algorithm=
    "http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
  <SignatureMethod Algorithm=
    "http://www.w3.org/2000/09/xmldsig\#rsa-sha1"/>
  <Reference URI="\#SAMLAssertion">
   <DigestMethod Algorithm=
    "http://www.w3.org/2000/07/xmldsig\#sha1"/>
   <DigestValue>
    (base 64 encoded sha1 hash of authentication token)
   </DigestValue>
  </Reference>
  <Reference URI="\#PaymentAssertion"
    Type="http://www.w3.org/2000/09/xmldsig\#SignatureProperties">
   <DigestMethod Algorithm=
    "http://www.w3.org/2000/07/xmldsig\#sha1"/>
   <DigestValue>
    (base 64 encoded sha1 hash of payment token)
   </DigestValue>
  </Reference>
 </SignedInfo>
 <SignatureValue>
   (base 64 encoded of rsa signature)
 </SignatureValue>
 <dsig:Object Id="SAMLAssertion" xmlns=""
    xmlns:dsig ="http://www.w3.org/2000/09/xmldsig\#">
  <AuthenticationAssertion AssertionID="SAMLAssertion"
    IssueInstant="(timestamp of authentication)"
    Issuer="(authentication authority)"
    xmnls="http://www.oasis-open.org/comittees/security/docs/
      draft-sstc-schema-assertion-15.xsd">
   <Subject>
    <NameIdentifier>
```

```
     <SecurityDomain>(Mobile Operator domain)</SecurityDomain>
     <Name>(base 64 encoding of user ID)</Name>
    </NameIdentifier>
   </Subject>
  </AuthenticationAssertion>
 </dsig:Object>
 <dsig:Object>
  <dsig:SignatureProperties>
   <dsig:SignatureProperty Id="PaymentAssertion">
    <app:Payment>
     <app:Supplier>(Web Service Provider URI)</app:Supplier>
     <app:Sku>(Web Service Item Code)</app:Sku>
     <app:Price>(Web Service Item Code Price)</app:Price>
    </app:Payment>
   </dsigSignatureProperty>
  </dsig:SignatureProperties>
 <dsig:Object>
</Signature>
```

B. Appendix 2

The Web Service Deployment Descriptor for the proof of concept `MatchCentreService` implementation is presented below:

```
<deployment xmlns="http://xml.apache.org/axis/wsdd/"
    xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">
 <service name="MatchCentreService"  style="rpc" use="encoded">
    <parameter name="className" value="com.madgo.simple.MatchCentreImpl"/>
    <parameter name="allowedMethods" value="*"/>
    <wsdlFile>/MatchCentreService.wsdl</wsdlFile>
    <beanMapping xmlns:ns="http://www.madgo.com/types/simple"
        qname="ns:ServiceConditions"
        languageSpecificType="java:com.madgo.simple.ServiceConditions"/>
    <beanMapping xmlns:ns="http://www.madgo.com/types/simple"
        qname="ns:CurrentScore"
        languageSpecificType="java:com.madgo.simple.CurrentScore"/>
    <beanMapping xmlns:ns="http://www.madgo.com/types/simple"
        qname="ns:MatchException"
        languageSpecificType="java:com.madgo.simple.MatchException"/>
 </service>
</deployment>
```

C. Appendix 3

A collapsed version of the Web Service Description Language (WSDL) for the proof of concept `MatchCentreService` implementation is presented below:

```
<?xml version="1.0" encoding="UTF-8" ?> - <wsdl:definitions
    targetNamespace="http://www.madgo.com/wsdl/simple"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:apachesoap="http://xml.apache.org/xml-soap"
    xmlns:impl="http://www.madgo.com/wsdl/simple"
    xmlns:intf="http://www.madgo.com/wsdl/simple"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:tns1="http://www.madgo.com/types/simple"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 + <wsdl:types>
 + <wsdl:message name="getAttendanceResponse">
 + <wsdl:message name="getAwayTeamResponse">
 + <wsdl:message name="getConditionsRequest">
 + <wsdl:message name="getConditionsResponse">
 + <wsdl:message name="getDurationResponse">
 + <wsdl:message name="getHomeTeamResponse">
 + <wsdl:message name="getScoreRequest">
 + <wsdl:message name="getImageResponse">
 + <wsdl:message name="getScoreResponse">
 + <wsdl:message name="getDurationRequest">
 + <wsdl:message name="getImageRequest">
 + <wsdl:message name="getAttendanceRequest">
 + <wsdl:message name="MatchException">
 + <wsdl:message name="getHomeTeamRequest">
 + <wsdl:message name="getAwayTeamRequest">
 + <wsdl:portType name="MatchCentre">
 - <wsdl:binding name="MatchCentrePortSoapBinding"
    type="impl:MatchCentre">
    <wsdlsoap:binding style="rpc"
        transport="http://schemas.xmlsoap.org/soap/http" />
  + <wsdl:operation name="getHomeTeam">
  + <wsdl:operation name="getAwayTeam">
  + <wsdl:operation name="getAttendance">
  + <wsdl:operation name="getDuration">
  + <wsdl:operation name="getScore">
  + <wsdl:operation name="getConditions">
  + <wsdl:operation name="getImage">
   </wsdl:binding>
-  <wsdl:service name="MatchCentreService">
  - <wsdl:port
    binding="impl:MatchCentrePortSoapBinding"
    name="MatchCentrePort">
     <wsdlsoap:address
        location="http://localhost:8080/matchcentre/
          services/MatchCentreService" />
    </wsdl:port>
   </wsdl:service>
 </wsdl:definitions>
```